ICANN86 Seville | PF – SSAC Lightning Talks (2 of 2)
Thursday, June 11, 2026 – 11:45 to 13:15 CEST

KATHY SCHNITT

Hello and welcome to the second of two SSAC Lightning Talk sessions. My name is Kathy and I'm the participation manager for this session. Please be advised that the session is being recorded and is governed by the ICANN Community Participant Code of Conduct, the ICANN Expected Standards of Behavior, and the ICANN Community Anti-Harassment Policy.

Regarding participation today, this session is designed for internal discussion among SSAC members on current security, stability, and resiliency topics. Observers are welcome to watch, but this session is not open for observer participation. For SSAC members, all members have been promoted to panelist status in Zoom.

To join the speaking queue, raise your hand feature, even if you are physically present in the room. When called upon, please state your name for the record. You may use the Zoom chat with one another, but please note that that is visible to observers in real time.

For those that are observers, the speaking queue is limited to SSAC members, and observer chat has been disabled for this session. With that, I'm happy to turn the floor back over to Barry Leiba.

BARRY LEIBA

Hi, this is Barry Leiba. Notwithstanding what Kathy just said, we will, if we have time, take questions from non-SSAC members, but the SSAC questions will take priority. If you're remote, type your question into the chat box, the Q&A box, sorry, and if you are in the room, then you'll use a floor mic or a desk mic to ask your question.

I doubt that we'll have time, but if we do, we will do that. Welcome to the second Lightning Talk session, my favorite part of the week. We have four truly exciting talks for you, so without further ado, Gustavo Ortega will take the first set.

GUSTAVO ORTEGA

Thank you. Let me share my presentation and then start. I hope everybody out there can hear because it is really loud up here with these fans. It's hard for me to hear. Okay. All right. Thank you. When AI Becomes the Attacker. So, this week we have been hearing a lot about AI. It's not hype. It's really changing the landscape.

And we should be raising up the concerns about what's happening, not just ICANN, but also you as individuals, you need to take this into your daily activities, professional activities, and raise up your concerns as well. So, we have here an igniting question. So, this igniting question basically is asking us if you know how many AI-driven attacks or DNS-driven attacks are already controlled by AI and we want to frame it under the AI Agentic behaviors.

So, raise your hand if you believe 100% of the DNS-based attacks are conducted through Agentic origins. You may already now know the answer, but are you watching this survey? Do you think it's 50%?

DANIELLE RUTHERFORD Can you please repeat the question because the acoustics are a bit bad? Thank you.

GUSTAVO ORTEGA Oh, you cannot hear me? So, the question that we have, it's what you can see, how many DNS driven attacks are conducted by Agentic AI? This is an igniting question. You just need to put it up front and say, well, if you believe it's 100%, raise your hand. If you believe it's 80%. Okay, we got one, 50, 60. I'm just throwing numbers, they don't have to be exact. 20 percent. Zero. Okay, we got some hands. So, keep that in mind, we will come back to this question later.

Now, before we move forward, it's important to scope what is Agentic AI. So, Agentic AI, and I make this distinction because it's important to differentiate it from the general AI. So, in this particular framework, I'm going to be referring to Agentic AI at that actuator in the internet. We have two sides of the coin. One is what we call the enemy, the attacker, and the other side is when we use AI for defenses to have basically security controls.

But from this perspective, I think it's important to make the distinction because sometimes we are just assuming that everything is the same. But no, the way how I conceive this and the way how I prefer it to be conceived is that we have AI as the oracle, that is the distribution of probabilities. And then we have an actuator that potentially can be malicious. And that's why basically cyber criminals are taking advantage of by using or relying on these Agentic properties and behaviors.

So, who am I? I'm Gustavo Ortega. I'm currently an SSAC member. It's an honor to be here as well as a fellow. And basically, what we're going to be covering today is divided across three main parts. So, the first part is Agentic AI. So, I like to see this from the eyes of a trend model. So, we're going to be basically identifying behaviors, characterizing these behaviors that we are going to be distinguishing whether they are classic or emergent.

And then part two, we are going to be discovering some of the potential defense mechanisms and emerging defense mechanisms. The last part basically it's an idea of why Agentic AI is basically so successful and basically, we're going to be discussing that down there to wrap up the session. So, it's important, especially, since we have an audience here to frame what's happening today in the landscape across the what we call the DNS stack.

So, there are some misconceptions already built in, in the way how the internet and the protocols were originated, how they were

constructed. Sometimes the misconceptions come from an anthropological perspective saying, well, it was not designed for security. It's not the case that it was not designed for security, it's just that the needs of those times didn't necessarily require to have the strength of controls that we have today.

But it is an evolution, so, we move from DNS, classic DNS, DNSSEC, and then in the process, we have services that needs to be from serving basically, the worldwide connection connectivity like the separate registration protocol, which is in this case, the main concern because that's what attackers and criminals are looking forward to compromise. Now, why is Agentic AI important?

Well, it matters because the differentiation comes at the speed of how things are happening. So, this is key because we need to concentrate and focus on the volumes of data that an Agentic entity can basically use. So, if we compare that at the levels of what we call human speed, definitely we see that we are orders of magnitude separated by this volume consumption.

So, while in a nutshell we can consume, let's say, a simple attacker individual maybe performing manual scripting, manual attacks, if he relies on the use of Agentic properties, he can achieve those orders of magnitude and submit, let's say, attack a particular target with plenty, high amount of requests.

And of course, the limitation there, well, there will be some infrastructure limitations like in, for instance, he might be taking down some of the resources like network interfaces because there

are physical limits. But of course, the amount of requests that can be processed simultaneously, especially if he's relying, let's say, on an orchestration that is using subproxies or sub-agentic properties, that will scale up.

So, let's move forward to part one. So, some of the emerging behaviors that we see are especially targeting, it's a mix, they're blended, so we will see behaviors that are emerging like using fabricated identities. These identities are not necessarily just identities in terms of what we call the human factor in terms of providing ways to identify itself, but also much at the machine level, at the infrastructure level.

The ability of Agentic AI to use some of the capabilities, for instance, through LLMs or other mechanisms to detect what are the patterns, what is the type of nomenclature or conventions needed to successfully create, let's say, transaction IDs. That's the classic example.

But of course, given the nature of AI itself, when it's relying on train models, that implies that basically that orchestration will learn with their failures. So, at least theoretically, we know that this is possible. And of course, something that I mentioned before is that capability of agentic AI to rely on the use of creating their own sub-agents to perform a series of specialized actions. So, that's what we call multi-agent swarming. So, all of that together helps to perform what we do call adaptive evasion.

So, basically, let's say well an agent can be orchestrated so that it will attack a single target, but it will hide its origin through multiple agents that are spread out worldwide although they will revert to the same origin. So, that's one way to execute the attack. Now we go back to fabricated identities as a threat itself.

So, the behavior that we see is that typically we are relying on mechanisms that, well, we have the ability to see metadata, so that metadata will tell us whether or not the Agentic AI basically, by consuming multiple volumes of data, will be able to recognize, like using pattern recognition or clustering to be able to generate the next transaction ID.

Of course, that's not the only one. There are other ways in how the Agentic AI will basically conduct these attacks. So, classic attacks like spoofing MAC identifiers, unfortunately, this is still happening today. And unfortunately, these settings, they are still being found in the wild. And since we don't have cryptographic identities, basically AI will come and say, well, I can consume a service and basically give it a fake MAC, and basically, potentially, if the vulnerability is identified, there will be a handshake.

So far, I have been mentioning about these threats of Agentic AI. So, this happens for a reason. Of course, what we need to consider here is the volume, but this volume happens because there are already known vulnerabilities, or simply they are not sufficiently strong to protect at the infrastructure level.

So, the classic threats like cache poisoning, DNS downgrade, song enumeration, they're still happening because simply, over the years, we have this technological depth, and basically, AI is coming with the ability not just to perform recognizance but also to have ways that will create payloads, and not just payloads, but will create and make the mechanism that orchestrate what we call the exploitation of this particular vulnerabilities.

So, AI basically will amplify what is existing and being at a different level, at a very distinctive level, the levels of criticality. That's what is really important here. It's not that AI magically will come in and idealize new patterns in the way how vulnerabilities are found. No, that's not the case. It's just that the use or the consumption of, let's say, knowledge to basically exploit like bits to be able to use known settings that is technically feasible to perform certain exploits.

Now the threat model from the perspective of how Agentic AI basically at front it's performing its activities is not that different from what we know today. It's however different in the sense of what it can do. Like for instance, it will still perform a recognizance of the infrastructure but as I was mentioning, the ability to use, let's say, knowledge, the classic perspective of going to the article that is AI per se and being able to identify, okay, what's happening if there is one bit here that is needed for security purpose but it's turned off.

So, that's what AI will basically be able to rely on and consume that to create payloads that can help basically escalate the level of

assertion over the different processes that are involved. Another example of another threat, it's a downgrade that will help us perform accelerated poisoning. So, it's not new, this is something that we already know, it has been happening now with AI.

But here the important aspect is that by consuming multiple volumes, it will not just help its own mechanisms to understand, let's say for instance, how transaction identifiers are being created, but also being able to perform simulations that, not necessarily simulations, but able to prove or to go and test some of the new conventions, no, some of the formats that are being created until some of them succeed.

So, something as simple as relying on this infrastructure, so we know we can compromise MAC identifiers, so that's just typically 48 bits, so it's not difficult for machine learning to come in and basically generate something that is useful according to what it observes.

So, if you see in the bottom, so the typical compromises, these are theoretical estimations of course, but let's say the attacker, it will take two hours versus the Agentic AI all the way to eight minutes. So, this is happening at a different volume, at a different level. It's important to understand that it's not that the AI itself is basically coming in and discovering necessarily other vulnerabilities.

I know there is a lot of commercial hype in the sense of what these AI models can do today, and that's why initially I frame it that we need to mention or separate, and this is philosophical. It's AI and

agentic behaviors. When we talk about something like Mythos, we are not just talking about AI as a knowledge convolution, but we are talking about an actuator, something that comes in the world and basically have interaction with the services, with what's out there to perform, to create payloads, to create scripting that will behave in the way criminals want it to work.

Now, this happens because, as I mentioned, there is a methodology. You may be already familiar with this methodology. This methodology is the kill chain process. It's not that different if we look at this from the perspective of what AI can do.

So, it will perform reconnaissance, it will perform weaponizing. That weaponizing is part of, it's a critical part of what it does in terms of relying on those discoveries of the potential vulnerabilities and tests by creating payloads that will attempt until one of them succeed, until there is some persisting in taking control of the target. For our purposes, that target is hijacking service registration protocols or being injected in the middle.

Now this is the most important slide that I have here because it really highlights what's happening in terms of the volume. So, volume is really what feeds AI, and agentic behaviors will basically become a differentiator because of this. So, keep that in mind that we as human, we have human factors, we are limited, but AI can work 24/7 every single day of the year.

And they can be replicated, they can be spammed, they can have activities, sub-processes that will target according to the principles

in computer science. It's this graph model. Then we have defending against AI. So, my time is limited here, but I will just mention this very briefly.

BARRY LEIBA Hit the highlights, you have two minutes. And that's including question time.

GUSTAVO ORTEGA Absolutely, thank you.

BARRY LEIBA Wrap it up.

GUSTAVO ORTEGA So, in terms of defenses, what we have today, we have very mature controls. We are also seeing patterns where there might be open-source frameworks that will come and use agentic AI to detect, let's say, when there is an exaltation of resources, when there is a spike coming in in terms of an attempt to hijack one resource. It's coming in. It's still in lapse.

So, probably by the end of this year, we will hear more about industries adopting these mechanisms. And the last part, basically, so the proof of concept that I mentioned, this is something that I have been working on my research as well. Not necessarily applied

to DNS, but it's basically the ability. So, I like to mix up concepts, like for instance, talking technically about philosophy.

So, for an agentic entity, the orchestrator basically will be able to conduct or to create multiple payloads until one of them succeeds. So, those are what we do manually, but instead it will have the capability to test all of them until one of them succeeds. So, finally going back to the original question, so all publications so far, they're saying that they all agree, from Kaspersky to Akamai and other sources, that fully combined with agentic, we don't see attacks, so it's less than 1%, but the landscape is changing.

And that's what I wanted to highlight today, which ICANN as a community, and SSAC in particular, we need to raise up the level of awareness if we want to continue to protect our businesses. Open to questions. Thank you.

BARRY LEIBA Okay, thanks. So, we have no time for questions on this segment. So, thanks, Gustavo. And next we have John Levine talking about how email uses the DNS.

JOHN LEVINE Amazing. It worked. Well, in many ways, this will be the exact opposite of the talk we just heard. We just heard about extremely new things in DNS attacks. Email, on the other hand, is extremely old. And that really is an important part of why it is the way it is. I mean, email predates the DNS, it even predates TCP IP.

I mean, RFC 771 was a discussion of how to move email from the old ARPANET NCP transport to this fancy new TCP IP that Vint and his friends had recently designed. Even after that, the email ran perfectly well with the traditional host.txt, which was simply a file of host names and IP addresses that was the manually distributed around the net, something that blockchains are attempting to recreate.

Then in RFC 974, and then we're still talking about the early 1980s here, we're talking about 40 years ago, explained how to use the DNS to find the hosts in email. So, in that era, I mean, the ARPANET and the internet was much smaller, it was much friendlier. I mean, so all of the traffic was unencrypted because everything was unencrypted.

All the other protocols people were using were unencrypted. We've attempted to add security stuff since then, but since email is such an important service, the updates have been done in ways that we tried really, really, really hard to make them backwards compatible.

So, if you implemented a mail server using RFC 821 from 1980, it would almost without any trouble at all, it would interoperate with mail servers that are implementing the latest standards that we're just working on now. And the way we've done this is simply by adding extra options to the baseline of simple unencrypted mail. Another thing that makes email rather peculiar is that it is not an interactive service.

You interact with your mail program, but it then hands stuff off, and messages are then passed around in the background by store and forward. So, the actual mail transfer entirely happens in the background. There is no human involvement. There is no reasonable way for like, if you have a website, you know, and it pops up a warning, you need to act on the warning.

There's nothing like that. It's all happened automatically. The mail invariably goes through multiple hops. It'll certainly go from a hop from your server to the recipient's server. It usually goes through several other servers on the way. That is both a bug and a feature. I mean, for example, one of my hosting providers forces all of the outgoing mail to go through their own local outgoing mail server.

So, we can do spam filtering, which is an important thing to do because hosting customers tend not to be very good about keeping their mail secure. We eventually added TLS, but the way we did it is by starting an unencrypted session and, as we'll see, moving up to an encrypted session. So, that's somewhat better, but, again, this was done a long time ago. The certificates were traditionally just local self-signed certificates.

So, there was no way that you could actually verify that the certificate matched the name of the server. So, you could still lie about who you are and mail could still be redirected. And without that, even in the most recent systems, mail is stored on the clear in each host that it passes through as it's relayed. And that's the

major security issue that people need to worry about for mail interception.

And in reality, it means we don't, because the main mail problems we worry about are spam and malware. I mean, the mail gets here fine. So, the question is, is it mail you want? Is it mail that you're willing to listen to? So, here is your basic description of the mail flow with a lot of annotations. We have the user at the upper left typing a message on her phone or her laptop or something.

And these days, it's more likely to be web mail. But that doesn't really make any difference. It then is passed to the mail server for her system, we call that process submission, which is similar to SMTP but slightly different. Then the initial mail server then needs to figure out where the mail is going to go. And it does that. It then uses SMTP to transfer the message from her mail server to the recipient's mail server. And again, there may be some other hops in between.

Once it arrives on the recipient mail server, then the recipient down there at the lower right, uses a different service called POP or IMAP to retrieve his message from the server. So, even in the simplest case, this has one, two, three different hops. And so, what are all the lookups for? And I'm not going to go through all these because I'm going to go through them in more detail. But the initial client needs to find the outgoing mail server.

The outgoing mail server needs to find the incoming mail server. The incoming mail server needs to do lots of checks on the mail to

figure out, is this mail we want to get? And then finally, the recipient needs to find his mail server to pick up his mail. So, here's the first set of DNS transactions, which is when you initially set up your mail program. And again, this doesn't apply to web mail, but it applies if you set up mail on your phone or if you set up mail on your laptop.

There's a certain amount of configuration that your mail program needs to do in order to handle the mail. It needs to know the IP address of the outgoing mail server. Since mail programs both send and receive mail if it picks up mail by POP or it picks up mail by IMAP, it needs to know where those servers are and those are typically different unless you have a very tiny mail system.

And it says here, there's too many different ways to set up mail. As often or not, it's manual, which means somebody simply says, here's the domain name of your mail server, and you type it into your mail program, and it then uses DNS-A or Quad-A lookups to make sure that that server exists.

There's also services called AutoDiscover, where you just give it your email address, and it consults a database of well-known mail systems and says, ah, you're on Gmail or you're on Yahoo, and here's the configurations you need.

A third way that is less common but actually works pretty well is there's serve records. Those are a DNS record that describes we're going to find a particular service. And you can see here there are six possible lookups. There's submission with and without built-in

TLS. There's IMAP and there's POP. And I publish all six of those for my users, and occasionally they're used. So, this is simply to set up your mail program. We haven't actually sent any mail yet.

So, here is the flow between the submission server and the and the initial mail server. And I go through this in a certain amount of detail just to show that there's all these interactions and there's a fair amount of DNS traffic required to make them work. So, typically, the user has written her mail message. She then pushes the send button. It then connects to the outgoing mail server and does a TLS negotiation at that point.

And then there's a sequence of the S colons are what the server says and the C is what the client says. So, the server says, here I am. The client says, hello. The server then responds with the extensions that it supports. And an important extension here is auth. I mean, yes, you can log in and say who you are, which ready is any sensible outgoing mail server. It won't send many mail unless you do that. Then the client says off plane and then sends a set of base 64.

So, if you decode that, you will actually know one of the login passwords from one of my mail servers, but I'm not going to tell you which one. Then once the authorization succeeds, then you mail from what's the return address, receipt to what's the address you're going to send to, and there can be several of those. And then once those are accepted, then it sends the entire message. The

headers in the body is one big block of mail, and then it says quit, and it's done.

So, for the submission, her mail server needs to do the A or Quad A lookups to find the outgoing mail server. Since it presents a TLS certificate, if you have a reasonably cool mail program, and there are a few of those, that understands DNSSEC, then it can do a TLSA lookup to check that the certificate that it got from the server is in fact this is in fact the certificate that belongs in that server.

This is simply a way to make sure that your outgoing mail is not being intercepted. Not super common, but it happens a certain amount. Okay. Now, the outgoing server has the message. What's it going to do with it? The answer is, it's going to send it to the recipient server. So, it first needs to figure out where the recipient server is going to be, and then once it's done that, it needs to figure out am I actually talking to the correct server or am I being a man in the middle here?

So, finding the recipient server, you do an MX lookup. MX provides a level of direction. When you do an MX lookup, it actually returns a list of both domain names and priorities with lower priorities being better. And these are, in fact, ICANN's mail server. ICANN has four mail servers with names Petrora 1 through 1, 6, 7, and 8. They all have the same priority, which means that pick one at random.

So, it does this for load sharing. And then once you've done the MX lookup, then you need to go back and do an A or a quad A lookup to find out the actual IP address that you connect to it. One of the

short-term hacks that we did back in the 1980s is, well, MX records were new. So, that temporarily, if there's no MX record, but a host does have an A record, then you pretend there was an MX record with the same name as the A record.

And as is always the case in software, that was intended to be a short-term transition, and here we are 45 years later, and we're still doing the same thing. So, now we've found the outgoing mail server, and now we have the relay mail flow, which looks a lot like the submission mail flow. So, we do the MX of the Quad A lookup to find the recipient mail server, and it says, hello, here's who I am. The client says hello, and then it comes back with a list of possible extensions. One of the extensions this time is start TLS.

So, now it starts with a plain text session. Now the client says start TLS. The server says, okay. Then it does the TLS negotiation. This is the point at which the client can now try and figure out whether it's talking to the right recipient server.

Once that happens, then it restarts again. It does another hello. And then it does the same mail from receipt to and data. And then it transfers the entire message over to the recipient server. But as these arrows suggest, there's a whole bunch of DNS lookups happening in this process. So, the first one is, is this the right server? Or am I having traffic misdirected by a man-in-the-middle attack in between?

And traditionally, like we didn't care, the certificates were sort of random. The certificate names didn't necessarily match, I mean, it

was quite common that you would install a mail package and it would simply create a certificate with the name local host, and that was close enough. Now, everyone does a signed certificate and they get them signed. So, if you do DNSSEC, you can publish a TLSA record.

And this will actually be enforced. I mean, I can tell you from experience that when I first set up the TLSA records on my mail servers, I got one of them wrong. And it turned out that the mail system of my wife's mother's ISP was doing TLSA checks. So, the way I knew that I had my TLSAs broken is that my wife said, why can't my mother send me mail? Which is good. The security was working.

So, I fixed the TLSA. But it's important to keep in mind this is entirely voluntary. I published the TLSA. There's no requirement that anybody checks it. And in fact, even though a domain will publish a TLS, it will certainly hope that you use start TLS. If you don't, it'll still accept your mail. That was too simple. If you've ever checked Google's or Gmail's DNS, they do not sign stuff with DNSSEC for reasons that I'll let them explain.

So, we came up with a different system called MTA-STS, which is supposed to provide a similar level of security, but without requiring DNSSEC. So, with MTA-STS, there is a text record you publish, which you can show here, this MTA-STS record that basically says, I do MTA-STS. And if the recipient system sees that

you've published such a record, then it does a regular web fetch of this URL, mtast.domainname.well known.

And then that is a file full of text that contains a list of the MXs that are supposed to accept mail for your domain. So, you can make sure that you're talking to one of those MXs and some flags about whether you want reports of failures and stuff like that. Again, even though this stuff is available, even though Gmail and Google publish MTA-STS, they hope you use .TLS. But if you ignore it and you send plain text mail, it's still voluntary.

So, now we've gotten to the point where the sending server has talked to the receiving server and it's ready to send the message. Now we're looking in the other direction, the receiving server says, do I want to accept this mail? And more often than not, the answer is no. I mean, our estimates are that 90% of the mail on the Internet is spam. So, the chances of this is a bad message are pretty good.

So, how does the receiving server attempt to figure out whether the sending server is sending mail at once? And there's two related questions it asks. One question is, is this message actually from the party it purports to be from since authenticated mail is marginally more likely to be legitimate.

And also, once it's authenticated, then you can set up a reputation for domains. You say, well, this domain sends good mail, and I'll accept it. That domain sends bad mail, and I won't accept it.

Do I do this on the right? Yeah, here we go. Okay, so the first thing you do is there are DNS block lists, which basically are lists of IP addresses that you probably don't want to accept mail from. So, here's an example of DNS checks at Spamhaus, which is the best-known block list provider. So, the receiving server simply looks up the IP address of the sending server and says, is this a bad server?

And if the answer is it's a bad server, depending on how you interpret it. In my case, I trust Spamhaus enough that if it says it's a bad sender, I simply direct the mail into a spam trap and I don't look at it at all. But as you can see, the values on the right, the different values are different reasons that it might be bad. They're a famous spammer. It's a network that shouldn't be sending mail. It's a retail ISP that probably shouldn't be sending mail.

And I notice here that most of these block lists use a freemium model. You can make a certain number of queries for free, but if you're a large provider and you're making hundreds of thousands of queries per day, you need to set up an account. And the way they do that is by encoding your account name into the query. So, if you look there down at the bottom, that highlighted yellow string is basically a shortened version of my account token.

So, when I make those DNS queries, Spamhaus can tell that those are from me and they can apply those to my account other than just some random stranger. So, now we've done the DNS block list checks. Okay, the sending IP is okay. Now what? The argument to the hello is supposed to be the domain name of the sending server.

You invariably do an A lookup on that. Any properly configured sending server will tell you its true name. And here, example, this is one of ICANN sending servers. You do the IP address check. You make sure it is the IP you want to hear from. Again, this is not specifically RFC required, but you want to do this because a lot of people will reject your mail if you don't. It's like, okay, now we know the server isn't lying about who it is.

The next thing is, how about the domain that the mail is being sent from? Same check. Is it a real domain that has a real MX or A record. If it doesn't, you almost certainly don't want it. And at the same way, there are DNS block lists for domain names. Check and make sure it's just a domain name that's problematic. In this case, ICANN seems to be okay, so we'll accept their mail.

Then we have an extraordinarily complicated scheme called SPF, Sender Policy Framework, where a domain says, these are the list of IP addresses that should be sending my mail. So, that you take the mail from domain address and you look up an SPF record, and this mess in the middle is in fact ICANN's SPF record. It's this long list of IP addresses. And the highlighted one happened to be the range that that message was being sent from.

But there's also an include thing. You say, well, it's all these things, and you include Salesforce like, well, you can also send mail from Salesforce. And this turns out to be the reason the SPF doesn't work very well is because so many organizations include

Salesforce, so many organizations say, well, we can also send stuff from Gmail or we can also send stuff from Outlook.

That it means that any Outlook customer can spoof mail from any other Outlook customer and it will still pass SPF. So, that ain't so useful. So, now we've gotten to the point where we're ready to look at the message. How much time do I have left?

DANIELLE RUTHERFORD Eight minutes.

JOHN LEVINE Okay, I'll do this a little faster. DKIM is a scheme where you can put a digital signature on the message with a lookup key in the DNS. So, again, on the right here, here's a DKIM signature. It has a cryptographic hash of the message. The recipient checks that the hash is right. Then on the left, there's a validation key. It can check and make sure that the signature on the hash is valid. And all that means is I handled this mail.

It doesn't mean it's good mail. It doesn't mean that it's nice mail. It just means that, again, if it's from ICANN, it's probably okay. If it's from other organizations, less so much so. There's another scheme called DMARC that sits on top of DKIM where a domain can say, all of my mail has a DKIM signature with my domain on it.

So, you do a DMARC lookup. In this case, it becomes from sales.example.com. You do a DMARC lookup for sales.example.com, and if there isn't one there, you walk up the

tree, okay, here's the DMARC record for example.com and it says what the policy is, which in this case is don't do anything.

And there's also some stuff in DMARC where you can send in aggregated reports so that other people can tell you what mail they think they're seeing from you. DMARC works really well for mail sent directly from a source to a reputation. It wreaks havoc on mailing lists because mailing lists will modify the mail and the DKIM signatures fail and we've been trying for years to fix that. One of the ways we've tried to fix it is something on ARC that I'm basically going to skip.

It was basically a way to say, well, the DMARC was okay before I modified the message, but that turns out not really to work. And then finally, for mail pickup, the recipient host logs in using Pop or IMAP to the recipient host and has to do an A lookup to find the host and might do a TLSA to check their certificate. So, there are a few odds and ends. It turns out you can have lots of MXs at different priorities.

So, here's an example where you have two round robins at the higher priority and then a backup server. Backup servers used to be common when the internet was flakier. Backup servers are not common at all now because it turns out that doing spam filtering on a backup server is really hard because the backup server never has the same information that the main server does. But the rotating multiple hosts is common.

And Yahoo, which is one of the big three mail servers, has three MXs, and each MX has six A records. So, basically, they're telling their senders to round robin through 18 different possible hosts. And the final thing is, one of the bad things about the A fallback is any domain that has an A record is potentially a mail host, which means that if you mistype something but the mistyped thing happened to be a domain with an A record, your sending server will keep trying for a week and failing to connect before your message would fail.

So, we fixed that. We invented something called Null MX. We can see MX0 on the host name is dot, which simply says no mail. So, it fails immediately. And that is it for the DNS stuff. So, each stage uses DNS to find and validate. The next stage, there's all sorts of security hacks layered on, like TLSA and MTA, STS, blah. And if we knew 45 years ago what we knew now, we would have designed it differently.

But it is a great triumph of the internet that this 45-year-old service still sits around billions of messages a day and we all use it. So, that's it. So, I think I have, what, two minutes? Three. Ooh. Questions?

BARRY LEIBA So, we have time for one or two questions. Danielle.

DANIELLE RUTHERFORD We got a hand from Peter Thomassen.

PETER THOMASSEN Hello. It's always interesting, I think, when services require use of domain names that don't really seem related to the service that is being used. For example, this auto-discover thing.

JOHN LEVINE Sorry?

PETER THOMASSEN The auto-discover mechanism.

JOHN LEVINE Yeah. If you're auto-discovering Yahoo or Gmail, it works great. If you're trying to auto-discover some tiny little domain like mine, it doesn't work at all.

PETER THOMASSEN Yeah. So, I wanted to point at a security risk that is there when you run services that give out subdomain names to users. So, actually, if you have a GitHub account, you get like your username.github.io, and you're used to get your username.github.com. So, once I renamed my username to autodiscover, and then, you know.

JOHN LEVINE Wow. Yes, you're right. There's a lot of band-aids. There have been a bunch of efforts to come up with a cleaner way to do autodiscover, but it always is boiled down to, well, it works well

enough for Thunderbird and webmail doesn't need it. So, it's never been worth fixing.

BARRY LEIBA Do we have one more? Dave.

DAVID LAWRENCE Hi, David Lawrence. I just wanted to provide one other piece of information on the problem you indicated with the TLSA records not actually working. On my personal domain, I also for a while had requiring TLS enabled, and that ended up unsubscribing me from several ancient mailing lists that I actually cared about because they were too old to bother doing TLS.

JOHN LEVINE Yeah, no, that is chronically a problem. It's like the guy who was revising the mail standard is one of the old timers, and his server is so old it only uses versions of TLS nobody will accept anymore. So, yeah, there's a lot of cruft there. TLS is nice and it does have the advantage that when it fails, you know that it fails. Once my wife complained, it took two minutes to fix. So, I think we'll be seeing more of that, but basically, it's limited by how much people use DNSSEC. And for the foreseeable future, that will still be not a whole lot.

BARRY LEIBA Okay. Thanks, John. And now we have Roberto Guerra.

ROBERT GUERRA

All righty. Good morning, everyone. This lightning talk hopefully will promote some conversation and dialogue afterwards. It's focused on cyber threats against the Tibetan diaspora. And what I'll say at the end, I'll say at the beginning, is how a community that's at risk community. All right. It's better now? No echo? Sorry about that. All righty. I'm just going to go.

I'm co-presenting this together with a colleague in Dharamsala who will speak about the group there. Next slide, please. So, why this talk and why today? This is not meant to be an advocacy talk. It's more a security briefing in terms of challenges being faced by a well-known target community that's being targeted of tons of DNS abuse, email abuse, targeted malware, and what they've done to try to document it, to collaborate, to teach, and to have that feedback loop to be safer.

And their interest really to collaborate with the community that's working with us at ICANN, SSAC as well. It's within SSAC's remit because we are working on issues of email, DNS abuse, and as well as IP addressing issues as well. And this talk will finish with perhaps some ideas in terms of the points of collaboration. Next slide.

A little bit about those who may not be familiar with the Tibetan community, without getting into too much of the geopolitics, there are about 100,000 Tibetans living in exile. And the TibCERT, which my colleague will be speaking about shortly, there are 50 organizations in Dharamsala and other parts as well that have been

collaborating together to try to document issues of Internet attacks, email issues and others.

They have been documenting issues of cyber targeting for the last 20 years. There used to be four hubs that were collecting information and doing training, but due to funding cuts, they're down to one. Next slide. Actually, go back if you can. Just some quick pictures here that are likely visible showing that there's presentations, awareness around passwords, trainings with different individuals, the issues of attachments and campaigns around detaching from attachments. And this has been taking place for the better part of 15 years or more. Next slide.

Just to give you a little bit of history. Again, there's a lot of information. I'm happy to follow up later. But showing that in this part of India, the internet arrived in 1997, and within two years, attacks started. Email started being spoofed. Viruses opposing to come from the Dalai Lama's office started attacking a variety of groups that work with His Holiness's office.

And in 2009, there was a very well-documented case of targeted malware called Tracking GhostNet, which was highly covered by the media, including the New York Times. This is well before a lot of the targeted malware attacks that we were getting a lot of reports years ago. So, it's a community that's been getting a lot of attacks.

I won't go through all of this and just an evolution of the type of attacks. Next slide. Kind of a quick summary. It's like from the

attacks have evolved over time from targeted malware to far more malicious extensions of the software that people are using. They're using big events, for example, and anniversaries and a lot of these attacks that are being focused on the Internet are now being seen in other ICT platforms.

So, there are now attacks occurring around cellular phones and the such. Next slide. Now, my colleague will talk the next couple of slides in terms of the Tibet Action Institute, TibCERT, and some of the attacks that they've been seeing. So, over to you.

TENZIN GYAL

Thank you, Robert. Am I audible? Okay, thank you. Hi everyone, for the record, I'm Tenzin Gyal, Program Manager at Tibet Action Institute. I've been working here for the past six years. I'll briefly talk about my organization. Tibet Action was established in 2009 by Lhadon Tethong and Nathan Freitas. We bring together the digital communication tools and strategic nonviolent actions to advance the Tibetan cause.

Our Director of Technology, Lobsang Gyatso Sither, is a well-respected voice in this community. He has testified before the U.S. Commission on International Religious Freedom and is well known for his advocacy work. Tibet Action has been recognized with several important honors for our impact, the 2019 Energy Democracy Award, the 2024 NED Democracy Award, and the 2024 Snow Lion Awards for the work that we have done on Colonial

Bodenschool. These awards reflect the seriousness of our mission and the real impact we're making.

Next slide, please. TibCERT stands for Tibetan Computer Emergency Readiness Team. It's a coalition-based program created by the Tibet Action Institute. Our mission is to protect the Tibetan community from targeted digital threats and surveillance and censorship. We serve as a dedicated cybersecurity response and research hub for the Tibetan in diaspora.

Also, it's important to note that Citizen Lab team supported us in the initial technical development and establishment of the TipCERT, thanks to Masashi and his entire team. So, here I'll briefly share some of our key engagements to TibCERT. First, research. We have published multiple reports working alongside security and research partners. Second, TibCERT infrastructure. We have built a proper help desk with dedicated incident response team. Third, finally we have training. We provide a lot of digital security trainings and help develop digital security policies for all of our members.

Next slid, please. So, this slide gives us a quick overview of the trends TibCERT has observed over the past two decades. Looking at the chart on the left, NGOs makes up nearly half of all the reported incidents. Tibetan activists are the second most targeted groups, followed by Central Tibetan Administration, CTA, and finally we have media groups.

Speaking of the dominant incident vectors, the largest category is the malicious email attachments, accounting for nearly half of all the incidents. This was one of the most common tactics used against Tibetan organizations for many years, still today, because it relied on more on trust and curiosity rather than technical vulnerabilities. The second most common attack factor is phishing. Then we have watering hole. This is a little different.

Instead of targeting individuals directly, attackers compromise the website that their intended victims frequently visit. And finally, we have these malicious links. So, compared to attachments, we have observed that links are often easier for attackers to distribute because they don't require sending a suspicious file. These findings are available in our reports, 2024 reports titled Cyber Espionage Against Tibetans.

What is striking about these attack vectors is that most of them rely heavily on human behavior rather than highly advanced hacking techniques. Although we have seen many sophisticated ones too. And most recently with the phishing campaign, we also have observed that attackers are trying to understand the system configuration of our organizations, many of the organizations, and also, they're trying to understand the response to these attacks.

And the most important takeaways is that the citizen left-finding at the bottom, their research showed that simply not opening unexpected email attachments could have prevented over 95% of the malware reaching. Thank you so much, Robert. Over to you.

ROBERT GUERRA

Great, thank you so much. Next slide. So, it's not just about attacks. So, the other thing too is in terms of attacks is also it is a community that has a different script. They're not using Latin characters, but they're Tibetan script has not yet appeared in the root, so they have to use other scripts. And because a variety of different type of characters and things like that can occur, the attackers are taking advantage of that.

And so, I will get into more details later with some sort of follow-up, but this is now a community that because it's not their first language, there can be a lot of attacks on lookalike or homoglyph domains. And so, one of the things that could be of interest is the TibCERT and Tibet Action Institute is very interested in working with those at ICANN and the larger community to get the Tibetan script into the zone.

And there is some progress, but some steps remain, and working with those in the linguistic community would be great. And I'll mention that a little bit soon. Next slide. Some other types of issues in terms of where there's some overlap with what ICANN, SSAC, and others are working on is DNS abuse. So, the role of registrars, what is their role? Can some information be shared in terms of type of attacks?

And does that echo kind of the group that's going on in regards to where phishing kits and others are being placed in terms of hosting providers. TibCERT has documented that IP ranges in Asia Pacific

and in Hong Kong that are frequently where those types of phishing kits are available. And there's been a well-documented abuse in regards to sending infrastructure that's being used, particularly AWS S3 and GoDaddy. workspace additional details in the reports. Next slide.

So, as was mentioned earlier, of the information that has been documented using the ability that they have at TibCERT is 90% is via email. It's the predominant vector. And what are the type of themes and emails that are tracked over time that potentially can alert people that a phishing attack of some kind or an email attack of some kind is taking place? There are several that are listed on the screen.

One that is coming up soon is the birthday of His Holiness the Dalai Lama. There is always a peak of attacks that occur leading up to his birthday on July 6th. And already, there is an increase of attacks that are starting to be seen, talking about campaigns or things that are taking place, a ceremony. People will go, oh, it's great, and there will be some malicious content in the email. So, being able to follow up on what John mentioned earlier, finding ways that authentication mechanisms can be done so the email can be seen as being authentic and not otherwise would be great.

Next slide. Won't go into this too much, but a variety of different type of malware, targeted malware has been seen ranging from backdoors, remote access, mobile and clusters as well.

Next slide. Attacks are not just email, they're sustained DDoS attacks, there's exploits, there's watering holes, and increasingly seen both coming at a cellular, a mobile level. That's outside of ICANN's remit, but wanted to mention that as well.

Next slide. What's new, and I mentioned this as well, some of the attacks are seeing at a cellular level, the anniversary and birthday of His Holiness, Dalai Lama are things that we are looking forward towards 2026.

Next slide. And so, a lot of this information was based off a site visit I made after the meeting in Mumbai and recognizing that there was an interest to work on IDN issues. Already, there's been reaching out to and engaging with SSAC members and those with the ICANN community, what's the state of the deployment of the Tibetan script, and there's conversations taking place with others in the ICANN community that know other type of providers, what are type of nonprofit or NGO programs that are available. Cloudflare has accelerated their onboarding to their program that they have. And a discussion with hosting providers that might be able to understand that attacks may take place and not drop them, what are the better hosting providers? So, that discussion is taking place.

Next slide. So, what are some particular asks? Is for an organization that's been working on email, that's been working on DNS abuse and other issues, and have produced reports, are there ways that they can be in the room? Can their data not just be used

as reports that are read but involved in some of the conversations, the policy discussions, some of the technical discussions and standards that are taking place?

If there are issues related to DNS abuse and email abuse, is there a way to brief registries and registrars to see how they might be helpful? Can we engage further in regards to, again, as I mentioned, IDN? And again, for the email authentication issues for a community that's particularly targeted by these attacks, how can that be improved?

Next slide. And again, for a community that's been working on these issues for a long time, how can they be in the room? Next slide. There's a lot of resources. The slides will be available to everyone.

Next slide. I'd like to thank both of you, and I think we have a couple minutes left for Q&A. So, I open the floor for questions and answers. Thank you.

BARRY LEIBA All right. Thanks, Robert. And we do have several minutes for some questions. So, Danielle, run the queue, please.

DANIELLE RUTHERFORD All right. I'm taking questions from SSAC members first. I'm not seeing any hands in the Zoom room from SSAC. Scanning. Looking. All right, no questions from SSAC members. Robert, do

you want to open up to general questions? All right. Any questions from the audience?

JOHN LEVINE

You are right that there is no Tibetan in the root, but there are actually Tibetan script rules for upwards of 50 contracted domains. You can do Tibetan second-level names in .com and in the various Indian-language versions of .com. And I realize that's second best, but since you can do that right now. Is that a path to making things better?

ROBERT GUERRA

I will defer to my colleague Tenzin Gyal to kind of answer that question. That was something that did come up. And again, it's part of the education component is, is that being deployed? Is that a bit of knowledge and advice known to that community that they could use that or not? So, Gyal, can you answer that at all?

TENZIN GYAL

Sorry, can you repeat the question?

ROBERT GUERRA

The question I think was John was making a comment that though the Tibetan script is not available at the root for the top-level domain, it is available at second level domain. And the question is, as has TibCERT or Tibet Action Institute registered second level

domains in the Tibetan script and .com and others that have it available.

TENZIN GYAL: Thank you. As far as I know, I don't think we have registered any of the second level domain in Tibetan. One thing we observed is that if we use Tibetan script in the second level domain, it converts into something, a bit of different letters. So, we're still confused how to do that properly. Yeah, thank you.

JOHN LEVINE: Thank you.

DANIELLE RUTHERFORD: We have a question from Gautam in the audience.

GUATAM AKIWATE: Hi, Robert, and Tenzin as well. So, in the slides, I noticed that three of the four hubs were scratched out. Do you have a sense of why? Why was that? Was that deliberate?

ROBERT GUERRA: I think Gyal will mention more, but from my awareness is that the funding from the U.S. government that helped support the four hubs, there has been a dramatic decline in funding that's available, and that's decreased not only activities to vet action in others, but also a variety of open-source projects and other type of projects as

well. And so, that's the consequence of that and which means there's now less data, there's now less data collection, but the activities continue.

DANIELLE RUTHERFOR Any other questions?

BARRY LEIBA Okay. Thanks again, Robert. And now we have Warren Kumari with a presentation that will undoubtedly involve a lot of cats.

WARREN KUMARI Sadly, and shockingly, there are almost no cats in this presentation, and I realize that that's going to break a long-running thing. But anyway, hi everyone. Okay, it is displaying. This is going to be a relatively short thing on what I'm calling Blinded Unique Identifiers, and I'm hoping we have lots of time at the end for lots of questions, because presumably there are many issues with this.

So, what am I actually talking about? We used to have this sort of information in WHOIS typed data, but for a bunch of reasons, we had to remove it around privacy and things like that. What Blinded Unique Identifiers is proposing is that we should have some sort of way to be able to represent a registrant with an identity that is not actually sort of reversible back to them.

And the utility of this would be in Associated Domain Checks across registrars or across different types of entities. So, basically, a way

that we can introduce something like this, not necessarily actually publicly published and available, but a way that you can do Associated Domain Checks with some sort of tag across different entities. So, this is going to be a different type of presentation to my normal ones.

It's more of a sort of like play/act. And this is Bob. Bob is an evil hacker guy. He likes doing cybercrime stuff. This is Harry. Harry is a cyber security person. He likes making the internet better, and he finds bad actors like Bob, goes along, tries to get their domains taken down. In the current model, we have Bob going off and registering a bunch of bad domains. Through Registrar A, he registers evil.com and bad.net and bank-phish.io and similar things.

And in this particular case, he's using his email address. Obviously, hackers don't always use the same email address, but they usually have some sort of content which stays or identify which stays the same across multiple registrars. Bob now goes off through a different registrar and registers a bunch of other equally bad or also bad phishing, spam, malware, et cetera domains. He's registered through registrar B And now Harry goes off and discovers one of Bob's bad domains.

He discovers that Registrar A, there's this bad domain, let's say it's evil.com. He contacts the Registrar and says, hey, this particular website seems bad. It's clearly hosting a phishing page. It looks like a Bank of America phishing page. You should probably look

and see if this registrant has any other domains associated. This is sort of the Associated Domains Check that's being proposed.

Registrar A has a look and sees that the same registrant went off and registered a bunch of other names. They have a quick look. They also seem to be bad, like they're posting malware, they're also phishing pages, et cetera. Registrar A takes down these domain names. That's great. Bob has lost some of his domains. However, he's only lost his domains through registrar A. He still has all of his bad domains registered through registrar B.

And the reason for that is there's no way to share who the registrant is currently between registrars. Like Harry doesn't know who the registrant was. This information is not made available to him. So, he can't go to registrar B and be like, you should have a look and see if this registrant is also doing bad stuff. So, yep, we took down some of Bob's bad domains through one registrar, but they still exist through a bunch of others.

So, in order to understand some of the bits of how this works, it's useful to have some background on cryptographic hashes and HMACs. A note for the tech people in the room, these are wildly, wildly, wildly oversimplified. Also, there are some bits where the AI that generated the image didn't do a brilliant job. But the 50,000-foot overview of a cryptographic hash is you provide an input. And this particular example, the input is bob@example.com.

And a hash generates a unique fingerprint of the input and it has the really nice attribute that you cannot take the output, which in

this case is 126, and reverse it back to the input. There's cryptographic stuff that protects that. Another interesting and important thing to know about hashes is any difference in the input will result in a very different hash output.

So, if anything's slightly different in the input, the output will be very different. Obviously, this particular example is wildly oversimplified, right? If the output is always a fixed length of three digits, there could be many things that match the same three digits. In the real world, a hash is a much longer output. And Barry's looking like I've grown an extra head. Okay, cool. The other thing that's worth understanding is HMACs, Hashed Method Authentication Codes.

This is basically the same thing as a hash, except that there is also a secret key as well. What you do is you take the secret key, you append it or prepend it to the input, and then you generate a hash of that. This has most of the same attributes as a hash. It's always the same fixed length output. It's infeasible to go from the output back to the input, but it has the additional attribute of if you don't know the secret key, you can't generate the HMAC output.

Hopefully that was understandable to people. I'm going through this quickly so that we have time for questions. So, we have some interesting building blocks and primitives. How does this help solve any of our issues? Well, we'll go back to have a look at the scenario. This is evil hacker Bob. He has registered evil.com through Registrar A, and he's used his email address or his credit

card or his address or his fax number or some bit of information in the registration.

Registrar A takes a hash of the identifying information. We're just using email addresses because it's an easy example. But they generate a hash, and they send it to a central service. This hash has the nice attribute that it's always the same length and you can't map back from the hash to what the input was. The central service generates what I'm calling a blinded unique identifier. It's just an HMAC of the hash. It sends it back to registrar A.

Registrar A stores that somewhere near the registration data. Evil hacker Bob registers a domain name through registrar B. Registrar B does the same thing. They send a hash of the various bits of identifying info, central service. The central service generates a blinded unique identifier. Because the hash input is the same, the blinded unique identifier will be the same. They send it back to registrar B.

A couple of things I'll reiterate about the blinded unique identifier, you can't unwrap the blinded unique identifier back to any of the original input info. All it is, is a unique tag. So, that's all cool and interesting. How does this help? Well, Harry finds one of Bob's bad domains. He contacts registrar A and says, this domain seems bad.

You should probably go off and look at all of the associated domains that the same registrant did. The registrar has a look and

sees, sure enough, this was a bad domain. Potentially, they find some other things that are also registered by the same registrar.

But now, because this doesn't really provide identifying information about the user, they can tell Harry, oh, the blinded unique identifier of this is this particular long string. Harry could then contact Registrar B and say, here's the blinded unique identifier. You should have a look and see if you see any registrations matching the same fingerprint.

Registrar B has a look and sees, yep, this particular blinded unique identifier also registered some other names and they also look like phishing. We should do something about that. And they can then do the takedown. Could also be like registrar A could just tell registrar B directly, we found some badness. Here's the fingerprint. Doesn't expose directly who the user is. It's a fingerprint that says this thing.

And now evil hacker Bob has lost all of his attack domains. Let me just check my time. Good, still have some time. So, some considerations in fact. Why are we suggesting that there's this complicated blinded unique identifier thing instead of just publishing the hash? Well, I said you can't reverse a hash back to its original input. But there is a thing called a dictionary attack.

Somebody could just try a.a.com, b.a.com, c.a.com, walk through all of those names, then try like different domain names, continuing to guess. So, instead, we take the hash and we create a blinded unique identifier. Well, if we have the central server, so

why don't we just ship like the raw data directly to them and be like, you guys just take care of this. Why do we say hash at first?

Well, you don't necessarily want to tell the central service, like, this is all of my registrant info. That seems like needlessly privacy-leaking stuff. I mean, the central service could do a dictionary attack because it now has the hashes. If it wanted to spend the CPU cycles, it could try and do a dictionary attack. Yeah, they could. Presumably, we're not going to treat the central service as being completely malicious.

Bob could change his email address and then the hash would be different. So, like if Bob made random characters in his email address uppercase, it could then result in a different hash. Yep, agreed. We could canonicalize the raw data first. And I can see, actually, I can't see John being ready to say the left-hand side is only the authority of it. But we could do some reasonable canonicalization stuff and get a reasonable heuristic.

Or if it's credit cards or phone numbers, we have ways of canonicalizing. I mean, Bob could just do stuff like being bob@example.com, bob1@example.com, bob@example.com and come up with different emails. Yep, that's true. Those will result in different blinded unique identifiers. But the proposal is instead of just doing email, do email, do the credit card number, do the phone number, do a bunch of other stuff which you generate these from.

Bob evil hacker could change all of the info in his registrant stuff. Yeah, they could. Those are no longer associated domains, right?

They're not actually connected through a registrant. It's a different registrar. I meant to prepare a bit before this, I just realized if Bob registers many different domains, like evil.com and phishing.net and a bunch of things, and a personal domain, and then on his personal domain, he puts his photo and his phone number, and somebody finds his personal address and some of the phishing ones, they now know it's the same person and Bob has exposed himself.

Well, yeah, I mean, yeah, these are all related to Bob. I guess if you're doing cybercrime, maybe don't put your LinkedIn profile on any of your attack domains would be a reasonable argument. These domains are associated. That's kind of a function of that.

Presumably at this point, a lot of folk are going, we've heard before we can build something at ICANN, but this is going to be wildly complicated and hugely expensive, and oh my God, it's going to be five years. It's really not that hard. I built a demo implementation. It took like an hour or two.

It would need a bit of more work before it gets productionized. But for the software engineers in the room, here is a 50,000-foot overview. You generate a hash. You ship it to a something. Something generates an HMAC, it ships it back, right? This is not complicated. For people who want more detail on the tech bit behind it, a thing that happens to ship back and forth is a protobuf. There's a request, there's a reply, there's a reply that contains the HMAC.

Again, this is not complicated. The actual bit that does any work is like three lines of code. And now, I was going to do a demo, but I actually forgot to get the demo set up. So, if you'll give me a second, I will quickly do that and I will open another window and I will go. There we go, that should be ready. Let me now make sure that I can share in here my screen. Share entire screen.

What could go wrong? Many things could go wrong. There we go. Is this actually working? So, as I say, I built a tiny demo. Here is the server part of this. Here is the client part of this. So, this would be the registrar type side. A registration comes in, test@example.com. It generates a hash. The hash gets sent to the central service, which is the other server part. The central service has replied with this blinded unique identifier. It helps if we're a new how to type.

If you enter the same. identifier, hash, phone number, whatever, you get back the same answer. If it's a different thing, you get back a completely different identifier. Unsurprisingly, this is not complicated. The demo also has a benchmark because doing it on one machine clearly is not actually reasonable, but the benchmark generates random emails in this case, ships them over to the server. Let's let this run for 15 seconds.

And just running on my laptop, which is not the world's fastest laptop, it does 113,000 of these a second. This was crappy code I wrote while watching a TV program. We could make this faster if needed. But 100,000 of these a second, I think, is more than fast

enough for what the proposed use case is. Let me now get back to. There was at least one cat. And questions. I think we still have four minutes.

DANIELL RUTHERFORD: All right. First, we're going to take questions from SSAC members. John, go ahead.

JOHN LEVINE: Technically, I think this is fine. I think, although it is morally indefensible to normalize email addresses. I think in practice, given that they all have to fit through registrar's forms, that's close enough, I'm worried about an attack from a malicious registrar.

WARREN KUMARI: So, yeah, if there are ways to mitigate the attackers, the registrar being malicious, an example would be when the registrar does a connection to the registry through EPP to do a registration, they could get back a redeemable token, which they then send to the service. And so, like we could limit the registrar's ability to.

JOHN LEVINE: Yeah, I'm also worried about, I mean, I don't know how realistic this is, but trying to reverse engineer contact information at other places so they can attack them. I wonder if this is this guy. I wonder if it's that guy. So, basically, not doing a full dictionary attack, but sort of doing a guided dictionary attack.

I guess what I'm saying is that could happen. I'm not sure whether it's enough of a risk that it's worth worrying about. But generally, the idea of, as a way to tie together, WHOIS information in different registrars without revealing it to the world. Yeah, it seems fine.

WARREN KUMARI: Yeah, and we should definitely chat more when I can actually hear you because the audio in this room is awful. But yeah, I guess that we should deal with malicious registrars. I think we can. Hopefully, there are more questions because --

DANIELLE RUTHERFORD: There are indeed. Rick Wilhelm, go ahead.

RICK WILHELM: Thank you; Rick Wilhelm. I don't have any questions. I do have a comment. I think that this is way overly rotated towards surveillance and enables a surveillance. I know I'm channeling others whose day job it is along this regard, but the kind of thing that this enables is really overly tilted towards observation and being able to connect people and their registrations across all TLDs in a manner that I think is inappropriate given the level of risk that's involved.

And Warren has a quizzical look on his face, but I don't think I'm misinterpreting what's going on here because we're talking about

one large pile of data, because also, I think that de-anonymization has been proven academically to be not really effective. Thank you.

WARREN KUMARI

So, if the blinded unique identifiers only shared once a domain has been shown to be maliciously registered, does that address a fair bit of your concern? And so, just like publishing all of these for anybody to get, if registrar A goes, this domain was malicious, we removed it, and it was clearly a bad registrant, and then it gets shared with other registrars, does that address or not?

RICK WILHELM

Deserves further study. I think that that ratchets it back, but I think that the long-term impacts of that would need to be contemplated. But I think that that deserves further study. I still think that needs to be considered. Thanks.

DANIELLE RUTHERFORD

All right. We've got less than 60 seconds left in the session, and I know that we still got some questions, but I'm going to ask the SSAC members to take those to the mailing list so that we can wrap up on time. Barry, over to you.

BARRY LEIBA

All right. Well, then I'll just say the last words of thank you to all the presenters. Thank you to everybody who came, and I hope you enjoyed it. Have a good rest of the last day.

[END OF TRANSCRIPTION]